

# Driveroo

Submitted by: Driverooers

Submitted to: Hao Lac  
ICET Department,  
School of Engineering Technology and Applied Science  
Progress Campus, Block A  
Centennial College

Discipline: Software Engineering / Game - Programming

Due Date: 2023-05-23

## **Declaration of Sole Authorship**

We, Driverooers, confirm that this work submitted for assessment is our own and is expressed in our own words. Any uses made within it of the works of any other author, in any form (ideas, equations, figures, texts, tables, programs), are properly acknowledged at the point of use. A list of the references used is included.

Signed:

- Muhammad Sabeeh, 301184564 (Game - Programming)
- Jin Huang, 301020707 (Game - Programming)
- Tannus Esquerdo, 301288342 (Software Eng Technology)

Date: 2023-05-23

## **Abstract**

The proposed project presents Driveroo, a mobile application that seeks to revolutionize the process of scheduling driving lessons by giving consumers a practical and tailored experience. Users of the app may search for driving instructors nearby, look at teacher profiles and reviews, set up appointments, and conveniently book lessons using the app. It responds to the rising desire for a quicker, easier way to locate driving instructors, particularly in regions with high demand.

The importance of Driveroo lies in its capacity to simplify the booking procedure, increase accessibility for students, and broaden the visibility of driving instructors. The software streamlines the scheduling and payment procedures, making it simpler for users to book courses. It also integrates payment channels. Additionally, teacher discussions guarantee a unique learning path for every user.

Potential users should download the Driveroo app and use it to look for driving instructors, make appointments for consultations, and reserve lessons in order to take full advantage of its features. Users may locate and book driving instructors in a more effective and easy manner by making use of the app's features and functions. The number of registered driving instructors, customer contentment, and favourable app store reviews will be used to gauge the project's success since they show the app's influence and worth within the driving instruction sector.

## **Table of Contents**

<b>Declaration of Sole Authorship</b>	<b>2</b>
	<b>3</b>

<b>Abstract</b>	<b>3</b>
<b>List of Figures</b>	<b>6</b>
<b>List of Tables</b>	<b>7</b>
<b>1.0 INTRODUCTION</b>	<b>8</b>
<b>2.0 METHODOLOGY AND RESULTS</b>	<b>9</b>
2.1 Literature Review	9
2.2 Proposed Solution	9
2.3 User Role Modelling	10
2.3.1 Brainstorm and Group	10
2.3.2 Consolidated User Roles	11
2.3.3 Description of User Roles and Persona	12
2.3.4 Additional Documentation	13
2.4 Release 1.0	14
2.4.1 User Stories	14
2.4.2 Additional Documentation	18
2.4.3 Release Plan 1.0	19
2.4.4 Iteration Plan (Release 1.0)	20
2.4.5 Additional Documentation	21
2.4.6 Progress Monitoring	22
2.4.7 Acceptance Tests for Release 1.0	23
2.5.1 User Stories	25
2.5.2 Additional Documentation	26
2.5.3 Release Plan 2.0	27
2.5.4 Iteration Plan (Release 2.0)	28
2.5.5 Additional Documentation	29
2.5.6 Progress Monitoring	30
2.5.7 Acceptance Tests for Release 2.0	31
<b>3.0 CONCLUSIONS</b>	<b>32</b>
<b>4.0 RECOMMENDATIONS</b>	<b>33</b>
<b>CREDITS, LICENSE, AND REFERENCES</b>	<b>34</b>
Credits	34
License	34
References	34
<b>APPENDIX A (DESIGN DOCUMENT)</b>	<b>35</b>

<b>APPENDIX B (TEST PLAN)</b>	<b>36</b>
1.0 Introduction	36
1.0.1 Goals	36
1.0.2 Assumptions	36
1.0.3 Risks And Assets	36
Describe the elements that might positively influence testing on the project.	
36	
2.0 Scope	36
2.0.1 Features To Be Tested	36
2.0.2 Features Not To Be Tested	36
3.0 Testing Procedures	37
3.0.1 Test Objectives	37
3.0.2 Types Of Testing	37
3.0.2.1 Unit Testing	37
3.0.2.2 Integration Testing	37
3.0.2.3 Acceptance Testing	37
3.0.2.4 Stress Testing	38
3.0.2.5 Performance Testing	38
3.0.3 Testing Tools	38
4.0 Schedule and Deliverables	38
<b>APPENDIX C (END-USER &amp; ADMINISTRATOR MANUALS)</b>	<b>40</b>

## List of Figures

Figure 1	Developer Perspective Diagram [1]	10
Figure 2	Organizing the user role cards on a table [1]	12
Figure 3	The consolidated role cards [1]	13
Figure 4	Consolidated User Low-Fidelity Prototype Diagram	15
Figure 5	Consolidated Instructor Low-Fidelity Prototype Diagram	16
Figure 6	UI Sketch Prototype For User Stories	44 - 64
<p>Figure #: The revised front of a story card with only the story and questions to be discussed [1].</p> <p>Figure #: Details that imply test cases are separated from the story itself. Here they are shown on the back of the story card [1].</p> <p>Figure #: An example of a constraint story card [1].</p> <p>Figure #: Possible electronic representation of a physical story card.</p> <p>Figure #: Iteration burndown chart for data from Table 5.</p>		

## List of Tables

Table 1	MoSCoW - Must-Have	26-27
Table 2	MoSCoW - Should-Have	28-29
Table 3	MoSCoW - Won't-Have	28-29
Table 4	Release Dates	26
Table 5	Release 1.0 Plan	29-30
Table 6.1	Disaggregated Tasks by Story: User Login Page	33
Table 6.2	Disaggregated Tasks by Story: User Home	34
Table 6.3	Disaggregated Tasks by Story: View Instructor Profile	35
Table 6.4	Disaggregated Tasks by	35

	Story: Booking	
Table 6.5	Disaggregated Tasks by Story: Profile	36
Table 6.6	Disaggregated Tasks by Story: Instructor Login	36
Table 6.7	Disaggregated Tasks by Story: Instructor Home	37
Table 6.8	Disaggregated Tasks by Story: Profile	38
Table 6.9	Disaggregated Tasks by Story: Booking Requests for Instructor	39
Table 6.10	Disaggregated Tasks by Story: View Detailed Booking	39
Table 6.11	Disaggregated Tasks by Story: Bookings For Instructor	40
Table 6.12	Disaggregated Tasks by Story: View Booking Details	40
<p>Table 1: The Must-Have stories for Release x.y [1].</p> <p>Table 2: The Should-Have stories for Release x.y [1].</p> <p>Table 3: Disaggregated tasks per story [1].</p> <p>Table 4: Stories, acceptance tests, and contributors for Release 1.0</p> <p>Table 5: Progress and changes for all iterations.</p>		

## **1.0 INTRODUCTION**

The technological issue addressed in this project is the slowness and absence of a simplified approach to booking driving lessons. The current techniques for locating and hiring driving instructors sometimes include time-consuming searches, restricted access to teacher information, and manual scheduling. The goal of this project is to create a mobile application called Driveroo to address this issue and provide a more effective way for learners to identify and book driving teachers.

This Technical Report (TR) describes the work done to propose and detail the development of the Driveroo app. The objective is to provide a user-friendly platform that streamlines the process of identifying and hiring driving instructors, enhancing the entire learning experience for students.

The report includes sections such as Introduction, Methodology and Results, Conclusions, Recommendations, and Appendices. It covers a literature review, a proposed solution, and user role modeling. The report focuses on planning, analysis, and user-centric aspects of the project, omitting specific implementation details. Procedures used include brainstorming, group work, user role modeling, release and iteration planning, progress monitoring, and acceptance testing. The scope encompasses the development of the Driveroo app and the processes involved in user role modeling, release planning, and iteration planning. It provides a comprehensive understanding of project objectives, user requirements, and the planning and testing processes involved in app development.



The primary goal is to create and introduce the Driveroo mobile application, which will let users look up driving instructors, review their profiles and reviews, schedule consultations, and book lessons all through the app. The emphasis on personalized experiences through talks with teachers prior to personalized scheduling of classes is what makes this concept distinctive.

## **2.0 METHODOLOGY AND RESULTS**

### **2.1 Literature Review**

Existing methods for identifying and booking driving instructors have shortcomings that Driveroo can overcome. These solutions are often tied to specific firms or driving schools, limiting independent teachers' access to new students. As a result, individual instructors have few possibilities to extend their student base beyond word-of-mouth recommendations and self-promotion. Furthermore, the existing process is inefficient and inconvenient. Users are frequently faced with a time-consuming process of picking appropriate packages, submitting required information, and being given random instructors without the chance to engage or discuss with them beforehand. Furthermore, scheduling is rigid and inflexible.

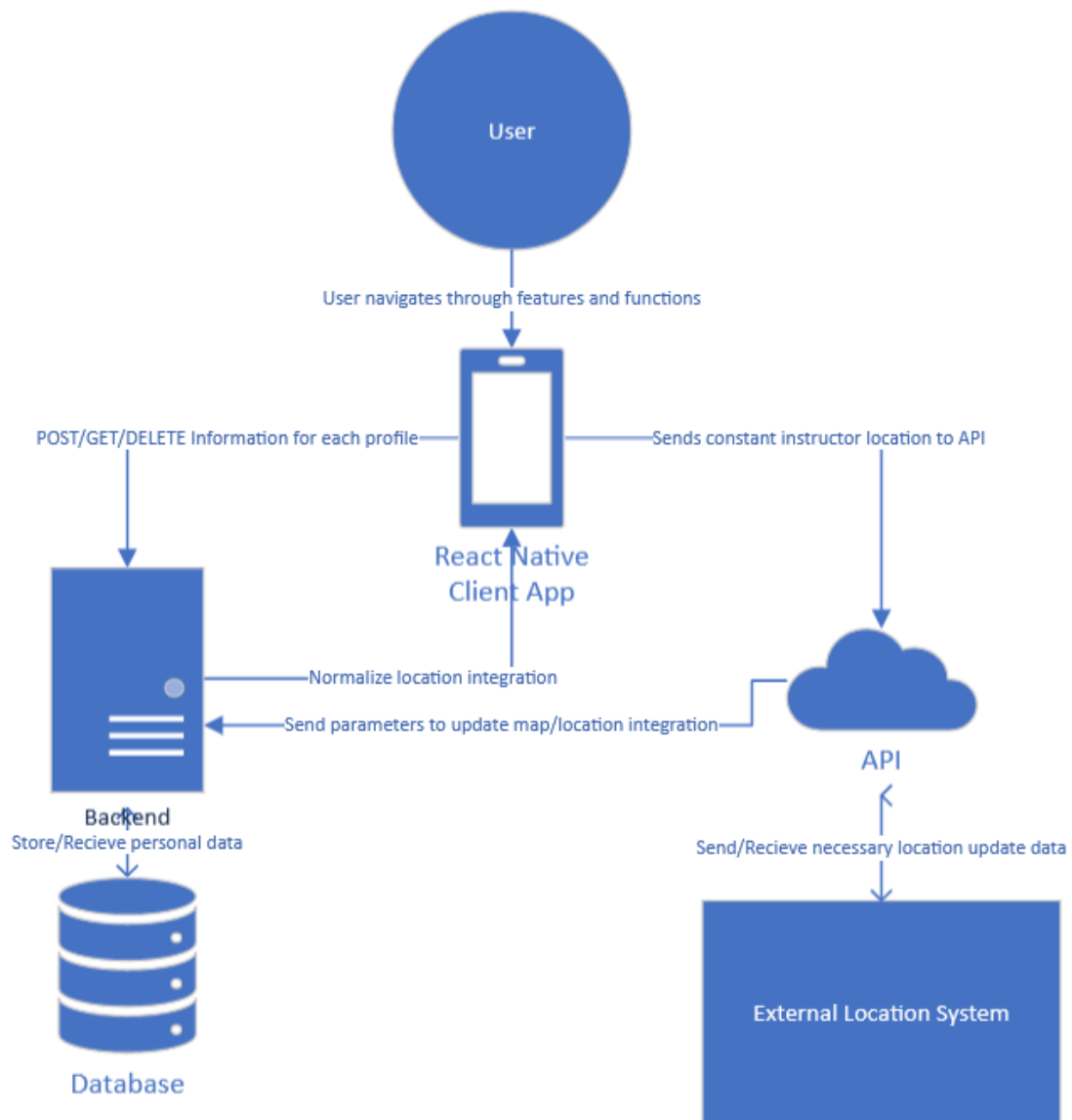
Despite these flaws, existing methods have some advantages, such as verification and recruitment processes. Driving school instructors are thoroughly vetted to ensure they have the required qualifications and credentials. Schools frequently hire instructors straight from graduating classes of teaching programs, resulting in a greater pool of trained teachers. These verification techniques instill

trust and professionalism in users by providing them with confidence in the teachers' abilities.

## **2.2 Proposed Solution**

Our system's strength to greatly ease the process of finding driving instructors for both consumers and employees is one of its main advantages. Users may easily access a wide network of driving instructors in their area by using our application, which is based on their general location. Users benefit from increased convenience and time and effort savings because of this functionality. Additionally, our platform provides a great level of flexibility, enabling customers to communicate openly with teachers about session scheduling and specific needs. The platform includes a strong profile matching system as well, allowing for easy pairing based on things like particular needs, instructor ratings, financial concerns, and schedule compatibility. This capability helps create a highly effective and customised experience, which ultimately optimises the user journey.

There are a few flaws in our system that need to be considered. First, it depends on the user's area having a sufficient number of driving instructors. If there aren't enough instructors, consumers may not be able to book courses through the app. Furthermore, the lengthy verification process for driving instructors that is required by regulatory rules. Although this procedure ensures compliance, it could cause delays when adding new teachers and growing the instructor network for the app. To provide a clear picture of the system's limitations, it is critical to identify and address these deficiencies.



**Figure 1: Developer Perspective Diagram**

[https://centennialcollegeedu-my.sharepoint.com/:u:/g/personal/jhuan145\\_my\\_centennialcollege\\_ca/EY4Pg4T08kxLg2\\_WFovnx3QBGQ2Qwz5mZ5ew3zRpoGFI8g?e=N6H3B0&action=edit&or=PrevEdit](https://centennialcollegeedu-my.sharepoint.com/:u:/g/personal/jhuan145_my_centennialcollege_ca/EY4Pg4T08kxLg2_WFovnx3QBGQ2Qwz5mZ5ew3zRpoGFI8g?e=N6H3B0&action=edit&or=PrevEdit)

## 2.3 User Role Modelling

### 2.3.1 Brainstorm and Group

During initial brainstorming, all user roles and groups were considered.

Therefore, the initial set can be considered overcomplete.

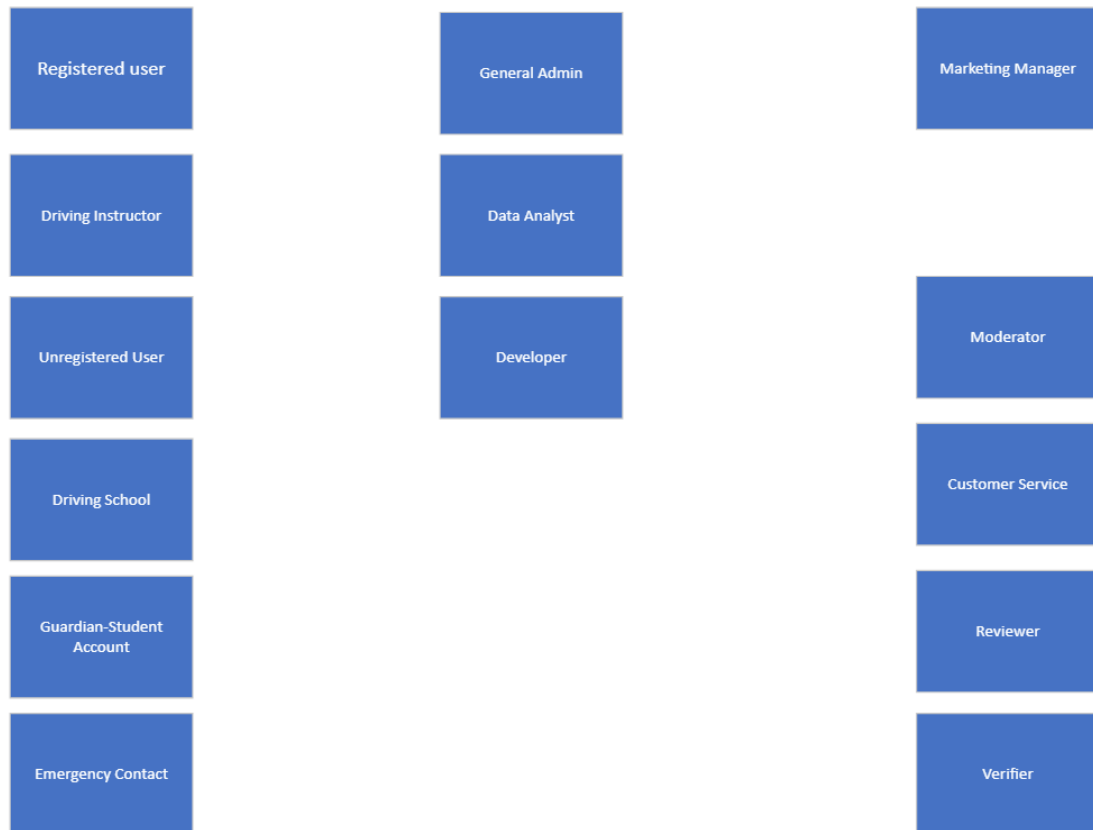
Following the brainstorming of the overcomplete set of roles, there were 4 clusters containing a total of 14 roles. These were organized by similar characteristics in context of our application

The figure for the initial set and organization of the set is properly captioned and accompanied by a discussion. The caption provides a clear and concise description of the content of the figure, accurately representing what is depicted, with any errors and issues fixed throughout the workshop. Additionally, the discussion accompanying the figure provides an analysis and explanation of the key elements and relationships within the set.

The number of user roles in the consolidated set is minimized; many roles that were considered similar due to their characteristics were either merged together or removed altogether, reducing the number from 14 to 6. Clusters were reorganized, which included the removal of unused clusters.

The consolidated set of user roles is properly captioned such that it provides a explanation of the role and all roles are grouped in clusters with each cluster properly labeled to provide a general view of the contents

The descriptions for each user role are discussed throughout the workshop, especially during the creation of the consolidated set.



**Figure 2: Organizing the user role cards on a table [1].**

### 2.3.2 Consolidated User Roles

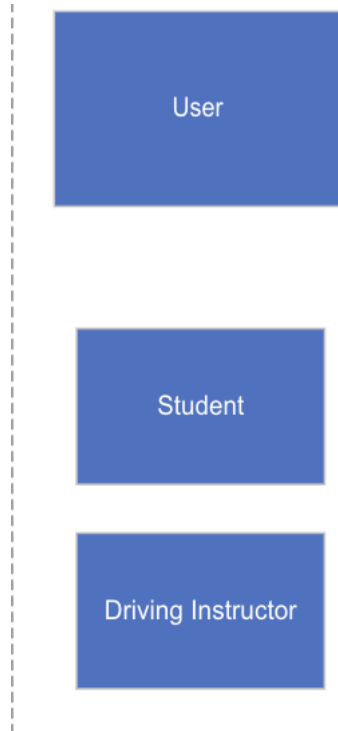


Figure 3: The consolidated role cards [1].

### 2.3.3 Description of User Roles and Persona

Role	User
Frequency	Somewhat Often
Expertise in domain	Dependent on the user
General Proficiency	Dependent on the user
Proficiency with Software	Dependent on the user
User's General goal	To book for and take driving lessons in order to gain appropriate knowledge for receiving their driving license

Role	Driving Instructor
Frequency	Very Often
Expertise in domain	Experienced

<b>General Proficiency</b>	Dependent on the user
<b>Proficiency with Software</b>	Experienced
<b>User's General goal</b>	To book students and give driving lessons/knowledge to help students receive driving license

### 2.3.4 Additional Documentation

Week 3 - User Roles video link:

<https://www.youtube.com/watch?v=jwZA95RWuP0>

## 2.4 Release 1.0

### 2.4.1 User Stories

During the workshop, the Driverooers team created low-fidelity prototypes for both user roles:

#### Consolidated Low Fidelity Diagrams

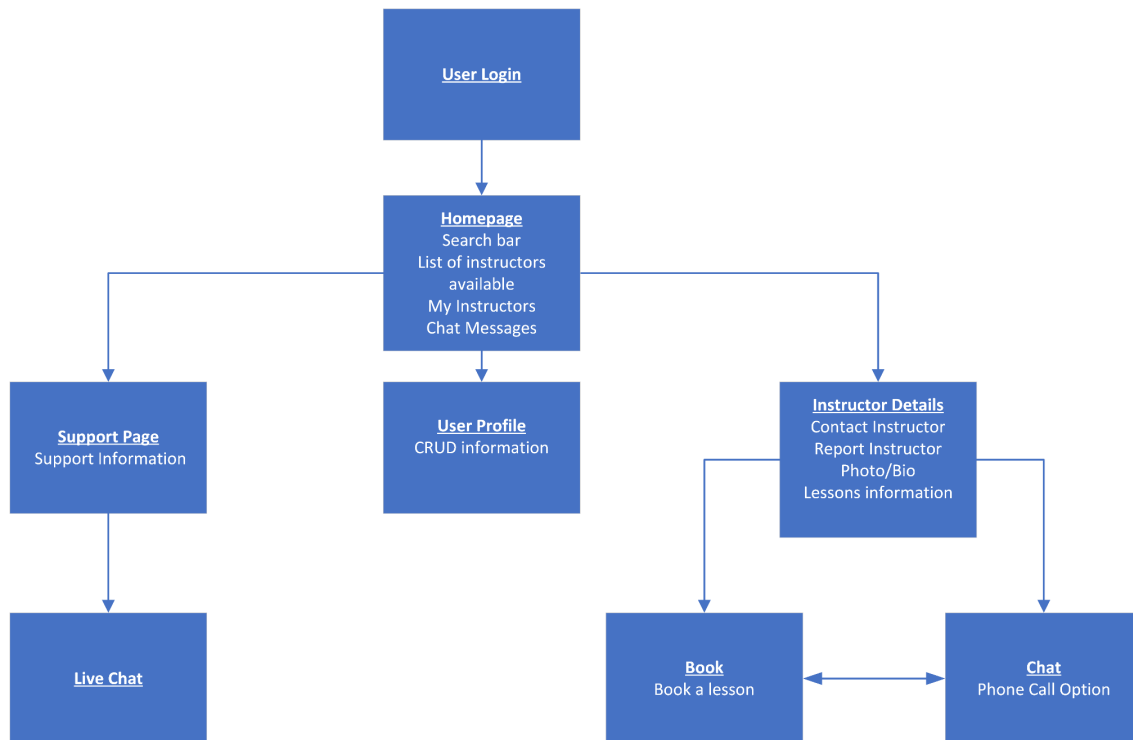
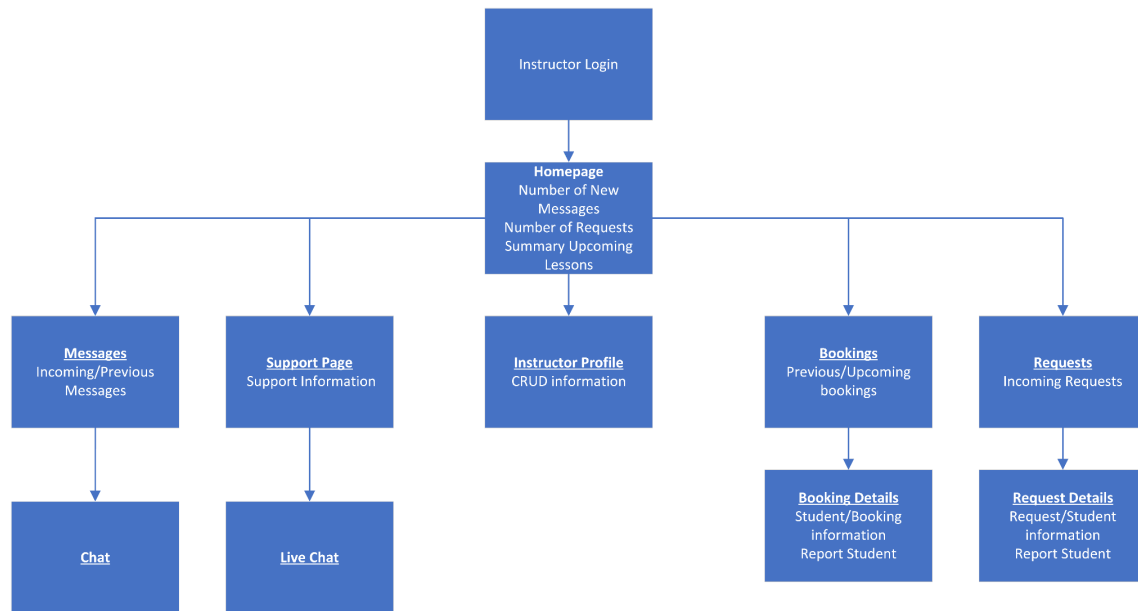


Figure 4 - Consolidated User Low-Fidelity Prototype Diagram





**Figure 5 - Consolidated Instructor Low-Fidelity Prototype Diagram**

The low-fidelity prototypes showcase a common progression for both user roles, requiring sign-in to access their respective homepages. Once signed in, users can update their personal profile information and utilize similar features, including messaging between different roles and customer service support.

Instructors, specifically, have access to their booking list and incoming requests for messages and bookings. They can also view detailed information about each booking and request, as well as engage in a chat box with users to discuss their specific needs.

Users, on the other hand, can browse available driving instructors' profiles and book lessons based on their availability. They can utilize the chat box to communicate with instructors and discuss their requirements during the lessons. Additionally, users can make online payments through the provided payment gateway to pay instructors in advance.

All user roles have access to a comprehensive homepage that offers an intuitive UI, enabling easy access to essential information such as messages, requests, and upcoming details.

For estimation purposes, one story point is equivalent to one hour of work for the development team.

A total of 20 User Story Cards were created and consolidated to cover the necessary functionality for the system.

<p>As a user, I want to be able to login to retrieve my data from the database (information, history, secure information, etc.)</p> <p><b>Estimated Time: 3 hrs</b></p>	<p><b>Acceptance Test:</b> test with fake user information <b>Expected Outcome:</b> result should show an error message saying invalid credentials, and tell you to sign up.</p> <p><b>Acceptance Test:</b> test with valid user information <b>Expected Outcome:</b> redirect to home page with content for logged in users now available</p>
---	--

<p>As a user, I want to be able to view a list of my instructors, along with a search bar/sort for filtering, and a page consisting of previous chat rooms</p> <p><b>Estimated Time: 36 hrs</b></p>	<p><b>Acceptance Test:</b> can find instructor via search bar <b>Expected Outcome:</b> result should consist of a list of the instructors with the same names and filter options</p> <p><b>Acceptance Test:</b> test user can view list of old/ongoing conversations and send messages <b>Expected Outcome:</b> conversation can ensue between instructor and test user</p>
---	---

<p>As a user, I want to be able to view details about the instructor, along a photo and short bio, allow for reporting, allow for contact with instructor, and have information about what will be taught</p> <p><b>Estimated Time: 12 hrs</b></p>	<p><b>Acceptance Test:</b> display details of a randomly selected instructor, and be able to engage in a chat with instructor</p> <p><b>Expected Outcome:</b> result should feature photo, bio, and chat option</p> <p><b>Acceptance Test:</b> test with reporting instructor</p> <p><b>Expected Outcome:</b> report should be sent to admins</p>
--	---

<p>As a user, I want to be able to chat with the instructor</p> <p><b>Estimated Time: 30 hrs</b></p>	<p><b>Acceptance Test:</b> test user can chat with the instructor and a phone call option should be working and available</p> <p><b>Expected Outcome:</b> test user and instructor can hold conversation via phone or chat messages</p>
--	---

<p>As a user, I want to be able to book the instructors</p> <p><b>Estimated Time: 35 hrs</b></p>	<p><b>Acceptance Test:</b> test user can book and pay for a lesson/package</p> <p><b>Expected Outcome:</b> should add booking to their account and payment should go through</p>
--	--

<p>As a user, I want to be able to view/update my information, such as name, email, phone-number, payment information, etc.</p> <p><b>Estimated Time: 8 hrs</b></p>	<p><b>Acceptance Test:</b> test with a test user, with fake information</p> <p><b>Expected Outcome:</b> display correct information for the user retrieved from database</p> <p><b>Acceptance Test:</b> update test user information</p> <p><b>Expected Outcome:</b> information should be updated immediately.</p>
---	---

	<p><b>Acceptance Test:</b> Delete test information</p> <p><b>Expected Outcome:</b> information must be deleted both from app and database</p> <p><b>Acceptance Test:</b> Delete test account and try logging in with same details</p> <p><b>Expected Outcome:</b> Account should be deleted and user should be redirected to homepage</p>
--	---

<p>As a user, I want to be able to contact a support to help me with any issues</p> <p><b>Estimated Time: 2 hrs</b></p>	<p><b>Acceptance Test:</b> test user should be able to see correct contact information and chat</p> <p><b>Expected Outcome:</b> Display correct information and a button to initiate chat with customer service</p>
---	---

<p>As a user, I want to be able to hold a live-chat with customer service</p> <p><b>Estimated Time: 30 hrs</b></p>	<p><b>Acceptance Test:</b> test user can initiate conversation with a real-human being</p> <p><b>Expected Outcome:</b> conversation should ensue and issue should be resolved</p>
--	---

<p>As an instructor, I want to be able to login to retrieve my data from the database (information, history, secure info, etc.)</p> <p><b>Estimated Time: 3 hrs</b></p>	<p><b>Acceptance Test:</b> test with fake user information</p> <p><b>Expected Outcome:</b> result should show an error message saying invalid credentials, and tell you to sign up.</p> <p><b>Acceptance Test:</b> test with valid user information</p> <p><b>Expected Outcome:</b> redirect to home page with content for logged in users now available</p>
---	--

<p>As an instructor, I want to be able to view at a glance important information pertaining to my upcoming lessons, any requests and a section consisting of number of unread message/message requests</p> <p><b>Estimated Time: 14 hrs</b></p>	<p><b>Acceptance Test:</b> test with populating instructors requests with test requests  <b>Expected Outcome:</b> instructor should be able to receive updated badge notification containing number of requests</p> <p><b>Acceptance Test:</b> test with adding new lessons to schedule  <b>Expected Outcome:</b> lessons should be added to the weekly overview</p> <p><b>Acceptance Test:</b> instructor can view messages from users  <b>Expected Outcome:</b> instructor can reinitiate conversation with users</p>
---	---

<p>As an instructor, I want to be able to CRUD my information, such as name, bio, photo, email, phone-number, banking info, etc.</p> <p><b>Estimated Time: 8 hrs</b></p>	<p><b>Acceptance Test:</b> test with a test instructor, and fake information  <b>Expected Outcome:</b> display correct information for the instructor retrieved from database</p> <p><b>Acceptance Test:</b> update test instructor information  <b>Expected Outcome:</b> Information should be updated immediately.</p> <p><b>Acceptance Test:</b> Delete test information  <b>Expected Outcome:</b> information must be deleted both from app and database</p> <p><b>Acceptance Test:</b> Delete test account and try logging in with same details  <b>Expected Outcome:</b> Account should be deleted and user should be redirected to homepage</p>
--	--

<p>As an instructor, I want to be able to view previous and upcoming bookings</p> <p><b>Estimated Time: 15 hrs</b></p>	<p><b>Acceptance Test:</b> test with a prior booking, with test information  <b>Expected Outcome:</b> display time and date of the booking, including the name of student</p> <p><b>Acceptance Test:</b> test with a new booking, with test information  <b>Expected Outcome:</b> display time and date of the booking, including the name of student,</p>
--	--

<p>As an instructor, I want to be able to view details about any bookings or report the student</p> <p><b>Estimated Time: 11 hrs</b></p>	<p><b>Acceptance Test:</b> test with a booking, with test information  <b>Expected Outcome:</b> display time and date of the booking, including the name of student and any other necessary information</p> <p><b>Acceptance Test:</b> test with reporting student  <b>Expected Outcome:</b> report should be</p>
--	---

<p>As an instructor, I want to be able to view incoming requests for bookings</p> <p><b>Estimated Time: 15 hrs</b></p>	<p><b>Acceptance Test:</b> test with a request for booking, with test information  <b>Expected Outcome:</b> display overview, including time and date of the booking, and the name of student</p>
--	---

<p>As an instructor, I want to be able to view detailed information about a request or report the request/student</p> <p><b>Estimated Time: 23 hrs</b></p>	<p><b>Acceptance Test:</b> test with a request for booking, with test information  <b>Expected Outcome:</b> display detailed view including time and date of the booking, including the name of</p>
--	---

	<p>student, along with any other information</p> <p><b>Acceptance Test:</b> test with reporting student</p> <p><b>Expected Outcome:</b> report should be sent to the admin</p> <p><b>Acceptance Test:</b> test with accepting/declining request</p> <p><b>Expected Outcome:</b> request should be removed from the requests page and moved to booking</p>
--	---

<p>As an instructor, I want to be able to contact support to help me with any issues</p> <p><b>Estimated Time: 2 hrs</b></p>	<p><b>Acceptance Test:</b> test user should be able to see correct contact information and chat</p> <p><b>Expected Outcome:</b> Display correct information and a button to initiate chat with customer service</p>
--	---

<p>As an Instructor, I want to be able to hold a live-chat with customer service</p> <p><b>Estimated Time: 30 hrs</b></p>	<p><b>Acceptance Test:</b> test user can initiate conversation with a real-human being</p> <p><b>Expected Outcome:</b> conversation should ensue and issue should be resolved</p>
---	---

<p>As an Instructor, I want to be able to view any previous or incoming messages from students</p> <p><b>Estimated Time: 8 hrs</b></p>	<p><b>Acceptance Test:</b> test with receiving new message</p> <p><b>Expected Outcome:</b> instructor should receive message and go to chat page</p> <p><b>Acceptance Test:</b> test if old/read messages are added to list of messages</p>
--	---

	<b>Expected Outcome:</b> messages should be visible.
--	--

As an Instructor, I want to be able to chat with students  <b>Estimated Time: 30hrs</b>	<b>Acceptance Test:</b> test with sending and receiving messages to a student <b>Expected Outcome:</b> student and instructor should be able to engage in conversation
---	---

## 2.4.2 Additional Documentation

<https://youtu.be/DMKk48kpl9o>

## 2.4.3 Release Plan 1.0

### Features:

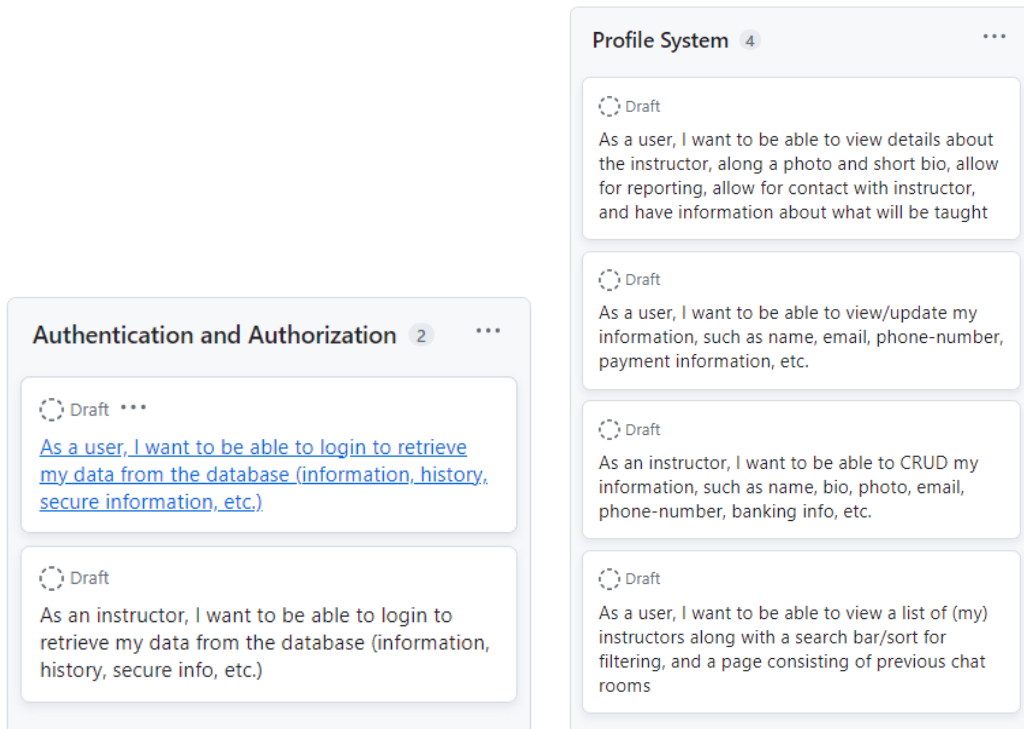
- Authentication and Authorization
- Profile System
- Booking System

### Secondary Features

- Chat Messaging
- Support



## Roadmap



Booking System6...

Draft

As a user, I want to be able to book the instructors

Draft

As an instructor, I want to be able to view previous and upcoming bookings

Draft

As an instructor, I want to be able to view details about any bookings or report the request/student

Draft

As an instructor, I want to be able to view incoming requests for bookings

Draft

As an instructor, I want to be able to view detailed information about a request or report the request/student

Draft

As an instructor, I want to be able to view at a glance important information pertaining to my upcoming lessons, any requests and a section consisting of number of unread message/message requests

Messages System3...

Draft

As an Instructor, I want to be able to chat with students

Draft

As a user, I want to be able to chat with the instructor

Draft

As an Instructor, I want to be able to view any previous or incoming messages from students

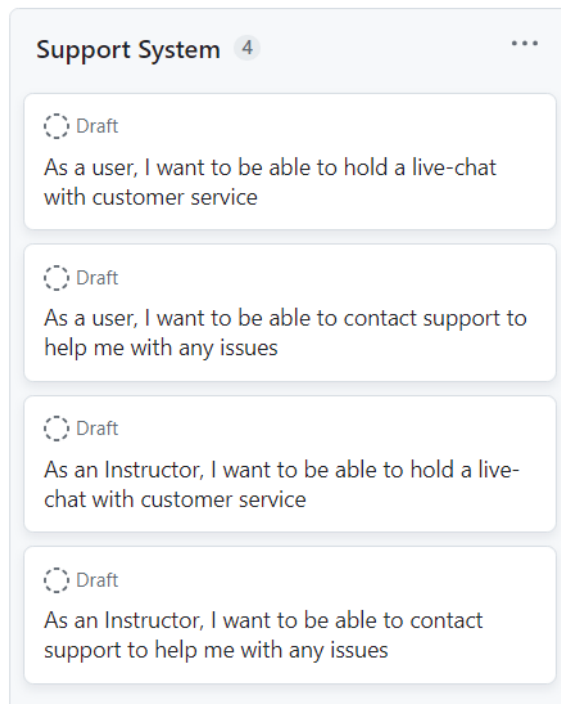


Figure 6: Project Development Roadmap

Name	Beginning Date	End Date
Iteration 1	July 9th, 2023	July 23rd, 2023
Iteration 2	July 23rd, 2023	August 6th, 2023
Release	August 13th, 2023	

Table 4: Release Dates

Must-Haves	
Story	Estimate
<b>Authentication and Authorization</b>	
As a user, I want to be able to login to retrieve my data from the database (information, history, secure information, etc.)	3
As an instructor, I want to be able to login to retrieve my data from the	3

database (information, history, secure info, etc.)	
<b>Profile System</b>	
As a user, I want to be able to view a list of (my) instructors along with a search bar/sort for filtering, and a page consisting of previous chat rooms	36
As a user, I want to be able to view details about the instructor, along a photo and short bio, allow for reporting, allow for contact with instructor, and have information about what will be taught	12
As an instructor, I want to be able to CRUD my information, such as name, bio, photo, email, phone-number, banking info, etc.	8
As a user, I want to be able to view/update my information, such as name, email, phone-number, payment information, etc.	8
<b>Booking System</b>	
As a user, I want to be able to book the instructors	35
As an instructor, I want to be able to view at a glance important information pertaining to my upcoming lessons, any requests and a section consisting of number of unread message/message requests	14
As an instructor, I want to be able to view previous and upcoming bookings	15
As an instructor, I want to be able to view details about any bookings or report the request/student	11
As an instructor, I want to be able to	15

view incoming requests for bookings	
As an instructor, I want to be able to view detailed information about a request or report the request/student	23

Table 1: The Must-Have stories for Release 1.0 [1].

Should-Haves	
Messages System	
Story	Estimate
As an Instructor, I want to be able to chat with students	30
As a user, I want to be able to chat with the instructor	30
As an Instructor, I want to be able to view any previous or incoming messages from students	8
Support System	
Story	Estimate
As a user, I want to be able to hold a live-chat with customer service	30
As a user, I want to be able to contact support to help me with any issues	2
As an Instructor, I want to be able to hold a live-chat with customer service	30
As an Instructor, I want to be able to contact support to help me with any issues	30

Table 2: The Should-Have stories for Release 1.0 [1]

Won't-Haves	
Story	Estimate
As admin, i want access to a	36

dashboard so that i can monitor reports on the app	
--	--

Table 3: The Won't-Have stories for Release 1.0 [1]

Iteration 1	Iteration 2
As a user, I want to be able to login to retrieve my data from the database (information, history, secure information, etc.)	As a user, I want to be able to contact support to help me with any issues
As an instructor, I want to be able to login to retrieve my data from the database (information, history, secure info, etc.)	As a user, I want to be able to chat with the instructor
As a user, I want to be able to view details about the instructor, along a photo and short bio, allow for reporting, allow for contact with instructor, and have information about what will be taught	As a user, I want to be able to hold a live-chat with customer service
As a user, I want to be able to view a list of (my) instructors along with a search bar/sort for filtering, and a page consisting of previous chat rooms	As a user, I want to be able to contact support to help me with any issues
As a user, I want to be able to book the instructors	As an Instructor, I want to be able to hold a live-chat with customer service
As a user, I want to be able to view/update my information, such as name, email, phone-number, payment information, etc.	As an Instructor, I want to be able to contact support to help me with any issues
As an instructor, I want to be able to CRUD my information, such as name, bio, photo, email, phone-number, banking info, etc.	As an Instructor, I want to be able to chat with students
As an instructor, I want to be able to view previous and upcoming bookings	As an Instructor, I want to be able to view any previous or incoming messages from students

As an instructor, I want to be able to view details about any bookings or report the request/student	
As an instructor, I want to be able to view incoming requests for bookings	
As an instructor, I want to be able to view detailed information about a request or report the request/student	
As an instructor, I want to be able to view at a glance important information pertaining to my upcoming lessons, any requests and a section consisting of number of unread message/message requests	

**Table 5: Release 1.0 Plan**

## 2.4.4 Iteration Plan (Release 1.0)

The following are required for this section:

1. Present each iteration plan with tables showing disaggregated tasks per story; a sample is shown in Table 3. See also the *Planning an Iteration* deliverable.
2. Discuss any discrepancies between the estimated and actual ideal time required to complete the tasks for the Table mentioned above.

Table 3: Disaggregated tasks per story [1].

Task	Who	Estimate	Actual
Code basic search screen	Susan	6	4
Code advanced search screen	Susan	8	9
Code results screen	Jay	6	8
Write and tune SQL to query the database for basic searches	Susan	4	3
Write and tune SQL to query the database for advanced searches	Susan	8	12
Document new functionality in help system and user's guide	Shannon	2	2



### Iteration 1 planning:

The base of the project (env variables, database connection) is already completed at this point

Pair Programmers:

- Jin, Guo
- Sabeeh, Tannus

As a user, I want to be able to login to retrieve my data from the database (information, history, secure information, etc.)			
Task	Who	Estimate	Actual
Backend			
Create a controller for role-based authentication	Krish	2	
Create a method to handle logging in	Jin	1	
Create a method to handle signing up	Guo	1	
Create a database model for the User	Zhiyang	2	
Frontend			
Create AuthContextProvider	Tannus	1	
Create Reducer function for AuthContextProvider	Tannus	1	
Create Login page and logic	Sabeeh	1	
Create Signup page and logic	Sabeeh	1	

Table 6.1: Disaggregated Tasks by Story: User Login Page

<b>As a user, I want to be able to view a list of my instructors, along with a search bar/sort for filtering, and a page consisting of previous chat rooms</b>			
<b>Task</b>	<b>Who</b>	<b>Estimate</b>	<b>Actual</b>
<b>Backend</b>			
Create the messages database model	Zhiyang	1	
Create the instructor database model	Zhiyang	1	
Create a method to send list of instructors based on token header	Jin	3	
Create a method to send data about active/old messages	Tannus	1	
Create a method to respond with list of instructors filtered by search term	Zhiyang	3	
<b>Frontend</b>			
Create non-functional homepage UI	Sabeeh	1	
Create function to handle logic for search bar	Tannus	1	
Create function to handle displaying list of my instructors	Sabeeh	1	
Create function to handle displaying most recent messages	Sabeeh	1	

Table 6.2: Disaggregated Tasks by Story: User Home

As a user, I want to be able to view details about the instructor, along a photo and short bio, allow for reporting, allow for contact with instructor, and have information about what will be taught			
Task	Who	Estimate	Actual
<b>Backend</b>			
Create endpoint to retrieve and send information about an instructor	Krish	3	
<b>Frontend</b>			
Create static UI for instructor details page	Sabeeh	1	
Implement logic to add to static pages based on instructor selection.	Tannus	1	

Table 6.3: Disaggregated Tasks by Story: View Instructor Profile

As a user, I want to be able to book the instructors			
Task	Who	Estimate	Actual
<b>Back End</b>			
Create endpoint to create a booking	Krish	3	
<b>Front End</b>			
Create scheduling UI (calendar)	Sabeeh	4	
Implement logic for booking	Tannus	2	

Table 6.4: Disaggregated Tasks by Story: Booking

<b>As a user, I want to be able to view/update my information, such as name, email, phone-number, payment information, etc.</b>			
<b>Task</b>	<b>Who</b>	<b>Estimate</b>	<b>Actual</b>
<b>Back End</b>			
Create an endpoint to retrieve/update and send user data	Krish	3	
<b>Front End</b>			
Create Static UI for profile page	Sabeeh	1	
Implement logic to display retrieved information	Sabeeh	1	

Table 6.5: Disaggregated Tasks by Story: Profile

<b>As an instructor, I want to be able to login to retrieve my data from the database (information, history, secure info, etc.)</b>			
<b>Task</b>	<b>Who</b>	<b>Estimate</b>	<b>Actual</b>
<b>Backend</b>			
Create a controller for role-based authentication	Krish	2	
Create a method to handle logging in	Jin	1	
Create a method to handle signing up	Guo	1	
Create a database model for the Instructor	Zhiyang	2	
<b>Frontend</b>			
Create AuthContextProvider	Tannus	1	
Create Reducer function for AuthContextProvider	Tannus	1	
Create Login page and logic	Sabeeh	1	
Create Signup page and logic	Sabeeh	1	

Table 6.6: Disaggregated Tasks by Story: Instructor Login

<b>As an instructor, I want to be able to view at a glance important information pertaining to my upcoming lessons, any requests and a section consisting of number of unread message/message requests</b>			
<b>Task</b>	<b>Who</b>	<b>Estimate</b>	<b>Actual</b>
<b>Backend</b>			
Create an endpoint and controller for sending data pertaining to story	Krish	4	
Create a method to send number of messages	Zhiyang	2	
Create a method to send number of requests	Zhiyang	2	
Create a method to send information about nearest bookings	Jin	2	
<b>Frontend</b>			
Create Static UI for instructor homepage	Sabeeh	1	
Implement logic for displaying and retrieving information	Sabeeh	2	

Table 6.7: Disaggregated Tasks by Story: Instructor Home

<b>As an instructor, I want to be able to CRUD my information, such as name, bio, photo, email, phone-number, banking info, etc.</b>			
<b>Task</b>	<b>Who</b>	<b>Estimate</b>	<b>Actual</b>
<b>Backend</b>			
Create endpoint and controller to update and send information	Krish	3	
Create method to update changed information	Zhiyang	2	
Create method to send instructor information	Zhiyang	2	
<b>Frontend</b>			
Create static UI for instructor information	Sabeeh	1	

Implement logic for displaying information	Tannus	2	
Implement logic for sending updates to endpoint	Tannus	2	

Table 6.8: Disaggregated Tasks by Story: Profile

<b>As an instructor, I want to be able to view incoming requests for bookings</b>			
<b>Task</b>	<b>Who</b>	<b>Estimate</b>	<b>Actual</b>
<b>Back End</b>			
Create endpoint and controller for bookings and requests	Krish	3	
Create method to send list of requests	Zhiyang	2	
<b>Front End</b>			
Create static UI for requests page	Sabeeh	1	
Display retrieved information on static page	Sabeeh	1	

Table 6.9: Disaggregated Tasks by Story: Booking Requests for Instructor

<b>As an instructor, I want to be able to view detailed information about a request or report the request/student</b>			
<b>Task</b>	<b>Who</b>	<b>Estimate</b>	<b>Actual</b>
<b>Back End</b>			
Create a method to send information about a selected request	Zhiyang	3	
<b>Front End</b>			
Create static UI for request detail page	Sabeeh	1	
Implement logic to retrieve and display data into static UI	Sabeeh	1	

Table 6.10: Disaggregated Tasks by Story: View Detailed Booking

As an instructor, I want to be able to view previous and upcoming bookings			
Task	Who	Estimate	Actual
<b>Back End</b>			
Create method to send list of bookings	Zhiyang	3	
<b>Front End</b>			
Create static ui for bookings page	Sabeeh	1	
Implement logic for retrieving and displaying bookings information	Tannus		

Table 6.11: Disaggregated Tasks by Story: Bookings For Instructor

As an instructor, I want to be able to view details about any bookings or report the booking/student			
Task	Who	Estimate	Actual
<b>Back End</b>			
Create a method to send information about a selected request	Zhiyang	3	
<b>Front End</b>			
Create static UI for request detail page	Sabeeh	1	
Implement logic to retrieve and display data into static UI	Sabeeh	1	

Table 6.12: Disaggregated Tasks by Story: View Booking Details

## 2.4.5 Additional Documentation

For this section, include 1 of 4 videos from your Iteration Planning meetings

(recall that you have a total of 4 Iteration Planning meetings)<sup>1</sup>:

1. Showing how your team disaggregated stories into their constituent tasks.
2. How developers on your team volunteer and take responsibilities for tasks.

Provide the file name and URL to the video(s) in your shared folder or YouTube channel.

---

<sup>1</sup> Indicate which iteration the video corresponds to. If you decide to submit a video in Release 1.0, then you do not need to include an *Additional Documentation* section for Release 2.0.



### **2.4.7 Acceptance Tests for Release 1.0**

The following are required for this section:

1. A table of stories and their associated acceptance tests for this Release as shown below in the sample in Table 5.
2. The link to your video demo for Release 1.0 stored either in a cloud drive, or your YouTube channel.

Table 4: Stories, acceptance tests, and contributors for Release 1.0 (Green=Passed; Red=Failed).

Full description of user story	Acceptance test(s)	Name(s) of contributing Developer(s)
As an User, I can ... so that .... <sup>2</sup>	Test with inputs .... Expected outcome: ...  Test with inputs .... Expected outcome: ...	Susan Smith, Jay Johnson
As an Administrator, I can ... so that .... <sup>3</sup>	Test with inputs .... Expected outcome: ...  Test with inputs .... Expected outcome: ...  Test with inputs .... Expected outcome: ...	Susan Smith, Jay Johnson, Shannon Shore, George Gavinson
As an User, I can ... so that ....	Test with inputs .... Expected outcome: ...  Test with inputs .... Expected outcome: ...  Test with inputs .... Expected outcome: ...	Jay Johnson, Shannon Shore, George Gavinson
As an User, I can ... so that .... <sup>4</sup>	Test with inputs .... Expected outcome: ...	Shannon Shore
As a Guest, I can ... so that ....	Test with inputs .... Expected outcome: ...  Test with inputs .... Expected outcome: ...  Test with inputs .... Expected outcome: ...	Susan Smith, Jay Johnson, Shannon Shore, George Gavinson, Abbey Appleby, Brian Bolt

<Insert url to video demo of Release 1.0 here>

<sup>2</sup> Green colour code indicates that all tests passed successfully as intended.

<sup>3</sup> Red colour code indicates that at least one test unintendedly failed.

<sup>4</sup> When all tests for a given story fails, this may suggest that implementation of the story has not even begun and indicates poor planning on the part of the team.

## **2.5 Release 2.0**

Release 2.0 has essentially the same structure as Release 1.0.

### **2.5.1 User Stories**

If your team wrote enough stories to cover up to or beyond Release 2.0 during your first story-writing workshop as described in the *User Stories* section 2.4.1, then your team will not need to hold a second formal workshop.

If a second workshop was held, submission for this section is the same as section 2.4.1.

### **2.5.2 Additional Documentation**

Include this section in your Technical Report only if your team required a second formal story-writing workshop. If a second workshop was held, submission for this section is the same as section 2.4.2.

### **2.5.3 Release Plan 2.0**

The requirements for this section are the same as section 2.4.3; update or add sections if required.

## **2.5.4 Iteration Plan (Release 2.0)**

The requirements for this section are the same as section 2.4.4.

### **2.5.5 Additional Documentation**

This section is required ONLY IF your team submitted materials for section 2.4.5.

### **2.5.7 Acceptance Tests for Release 2.0**

The requirements for this section follow the same requirements as in section 2.4.7 except acceptance testing is for stories allocated for Release 2.0 and incomplete stories subsequently moved from Release 1.0.



### **3.0 CONCLUSIONS**

A conclusion interprets the data found in the Body. It is reasoned judgment and not opinions. Consider the variables. Relate cause and effect. Analyze, evaluate, make comparisons and contrasts. Base the conclusion on fact.

## **4.0 RECOMMENDATIONS**

Recommendations are not required for all studies. They suggest a course of action and would generally be provided when there are additional areas for study, or if the reason for the TR was to determine the best action going forward.

## CREDITS, LICENSE, AND REFERENCES

### Credits

Provide any credits here. The following are examples:

Author of the template graphic layout : Hao Lac

<haolac.at.centennial@gmail.com>

Author of the template explanation text : John Doe

<[john.doe@centennialcollege.ca](mailto:john.doe@centennialcollege.ca)>

### License

State the license granted with your system. For example:

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the appendix entitled "GNU Free Documentation License".

### References


[1] Cohn, Mike. 2004. *User Stories Applied: For Agile Software Development*, Addison-Wesley Professional.

## APPENDIX A (DESIGN DOCUMENT)

Figure 6: UI Sketch Prototype For User Stories

As a user, I want to be able to login to retrieve my data from the database (information, history, secure information, etc.)

USER LOGIN



# Driveroo

## User Login

Username
Password
Submit

As a user, I want to be able to view a list of my instructors, along with a search bar/sort for filtering, and a page consisting of previous chat rooms

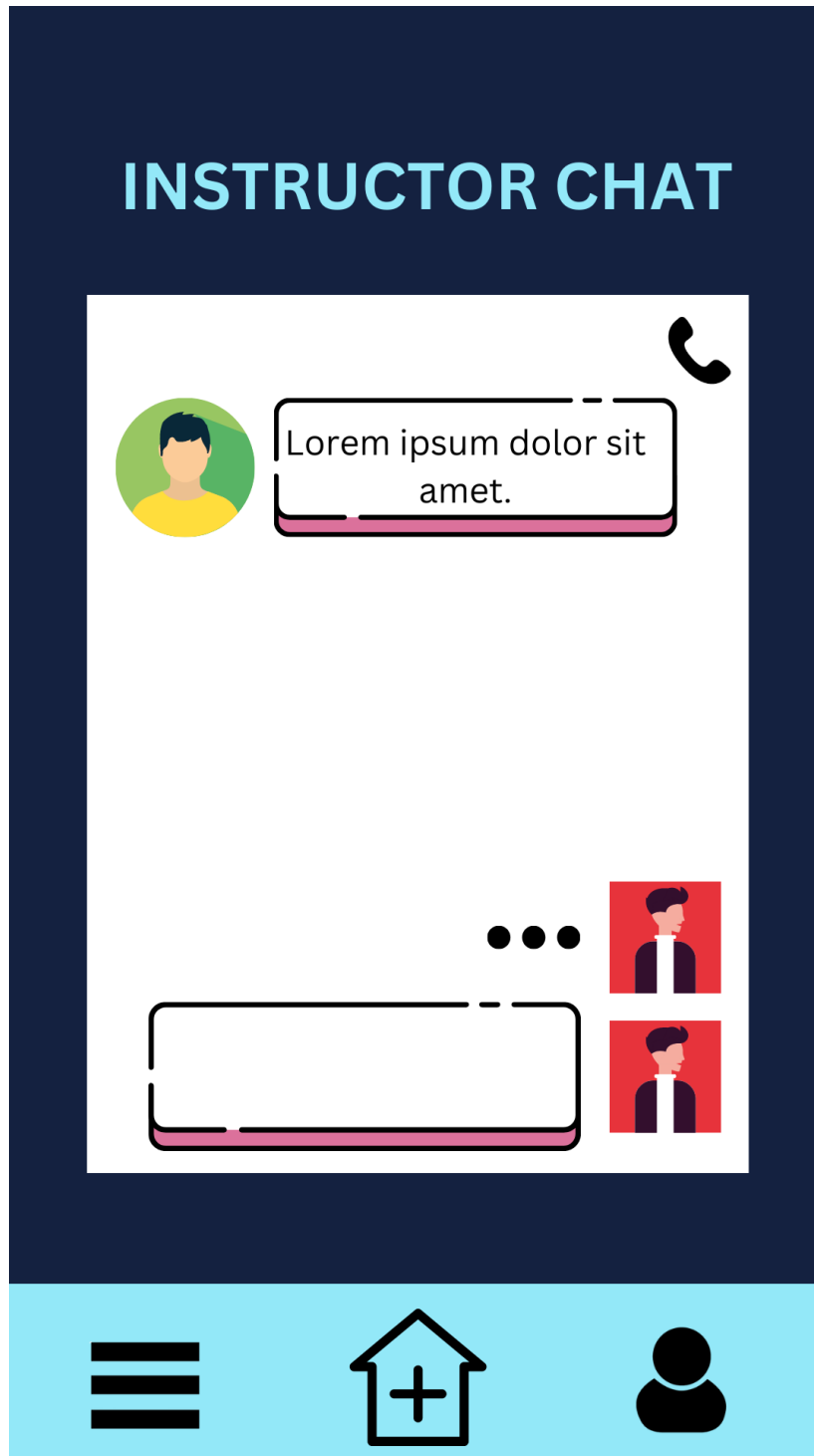


As a user, I want to be able to view details about the instructor, along a photo and short bio, allow for reporting, allow for contact with instructor, and have information about what will be taught




As a user, I want to be able to chat with the instructor

### User Chat




As a user, I want to be able to book the instructors

**INSTRUCTOR BOOKING**






# BOOKING



**Doe John**

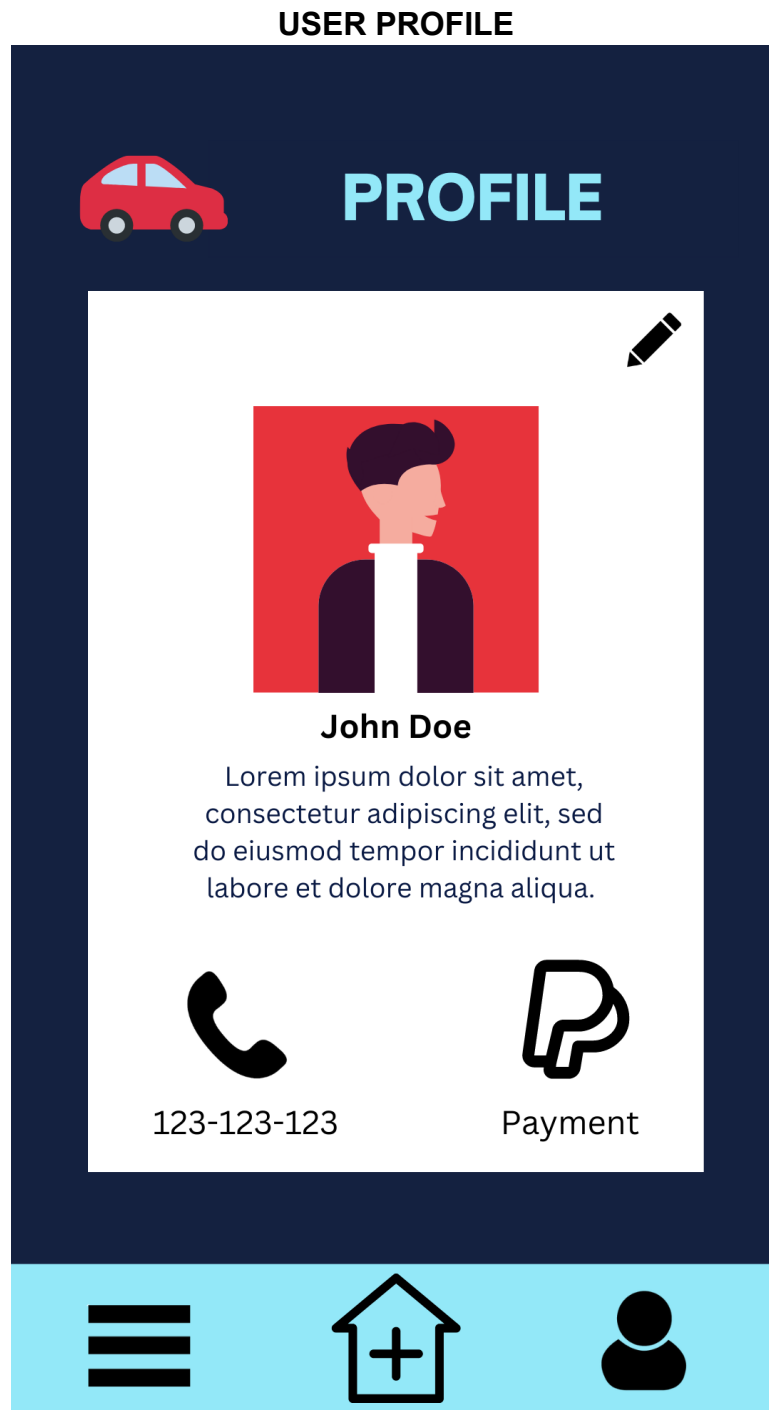
ThisDay	ThatDay
2:00pm	2:00pm
5:00pm	5:00pm

**Book**



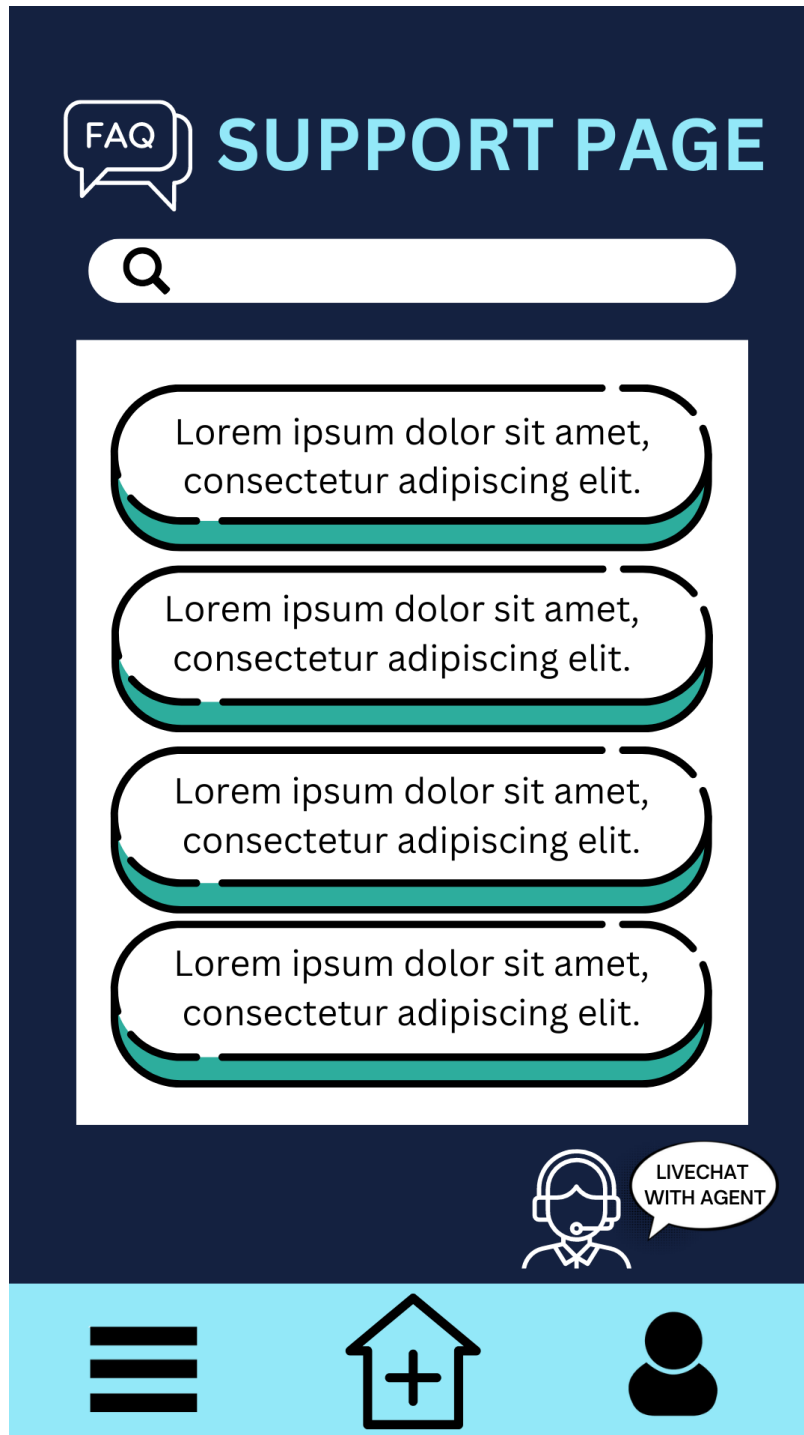


As a user, I want to be able to view/update my information, such as name, email, phone-number, payment information, etc.

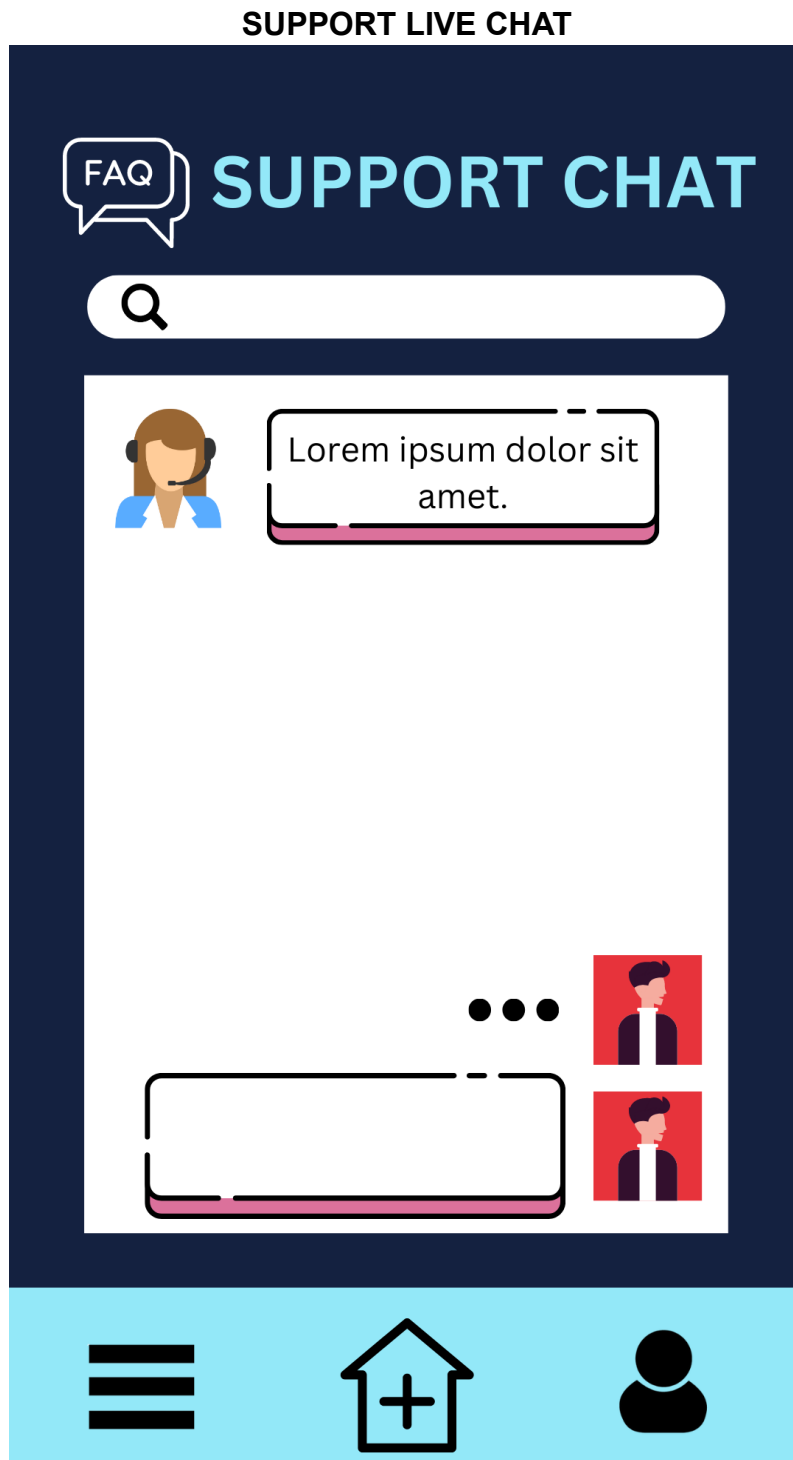


As a user, I want to be able to contact a support to help me with any issues

## SUPPORT PAGE



As a user, I want to be able to hold a live-chat with customer service



As an instructor, I want to be able to login to retrieve my data from the database (information, history, secure info, etc.)

## INSTRUCTOR LOGIN



# Driveroo

## Driver Login

As an instructor, I want to be able to view at a glance important information pertaining to my upcoming lessons, any requests and a section consisting of number of unread message/message requests



As an instructor, I want to be able to CRUD my information, such as name, bio, photo, email, phone-number, banking info, etc.



As an instructor, I want to be able to view incoming requests for bookings

## BOOKING REQUESTS (INSTRUCTOR)



As an instructor, I want to be able to view details about any bookings or report the student

### VIEW BOOKING DETAILS (INSTRUCTOR)





As an Instructor, I want to be able to view any previous or incoming messages from students

## STUDENT MESSAGES (INSTRUCTOR)



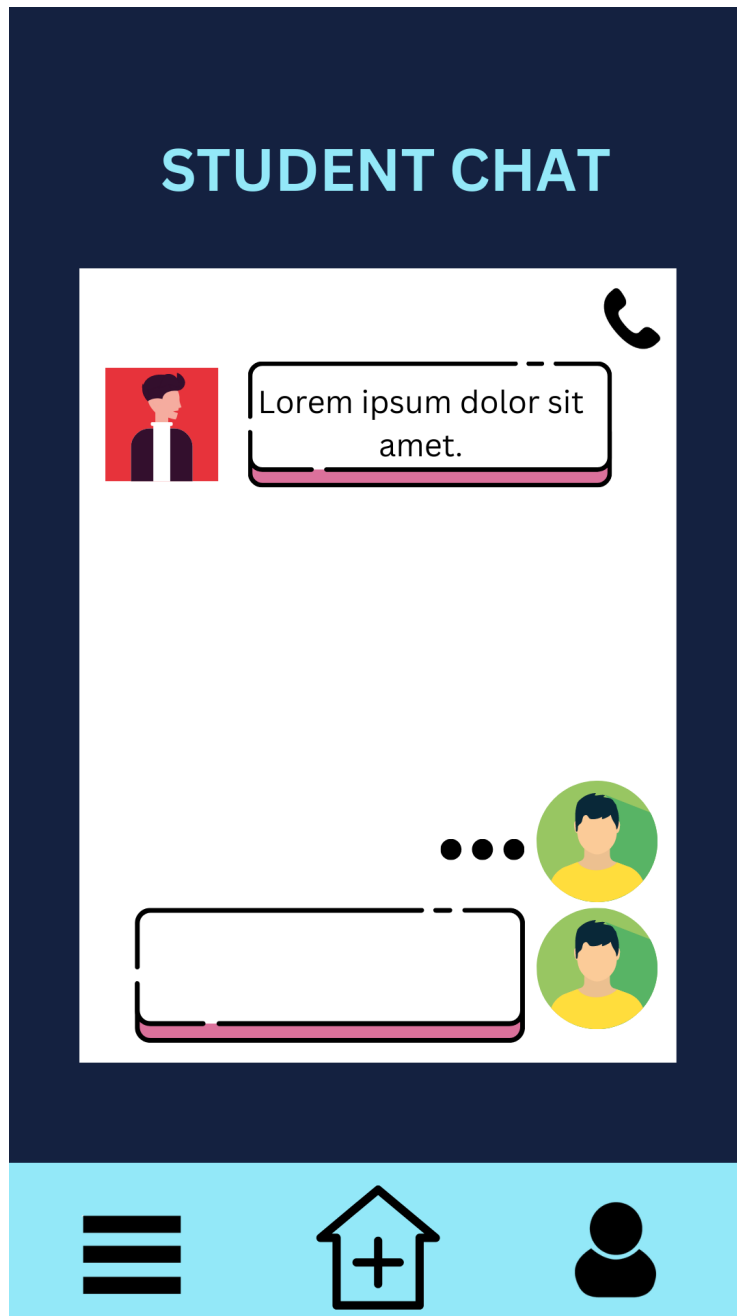
As an instructor, I want to be able to view detailed information about a request or report the request/student

## DETAILED INFORMATION



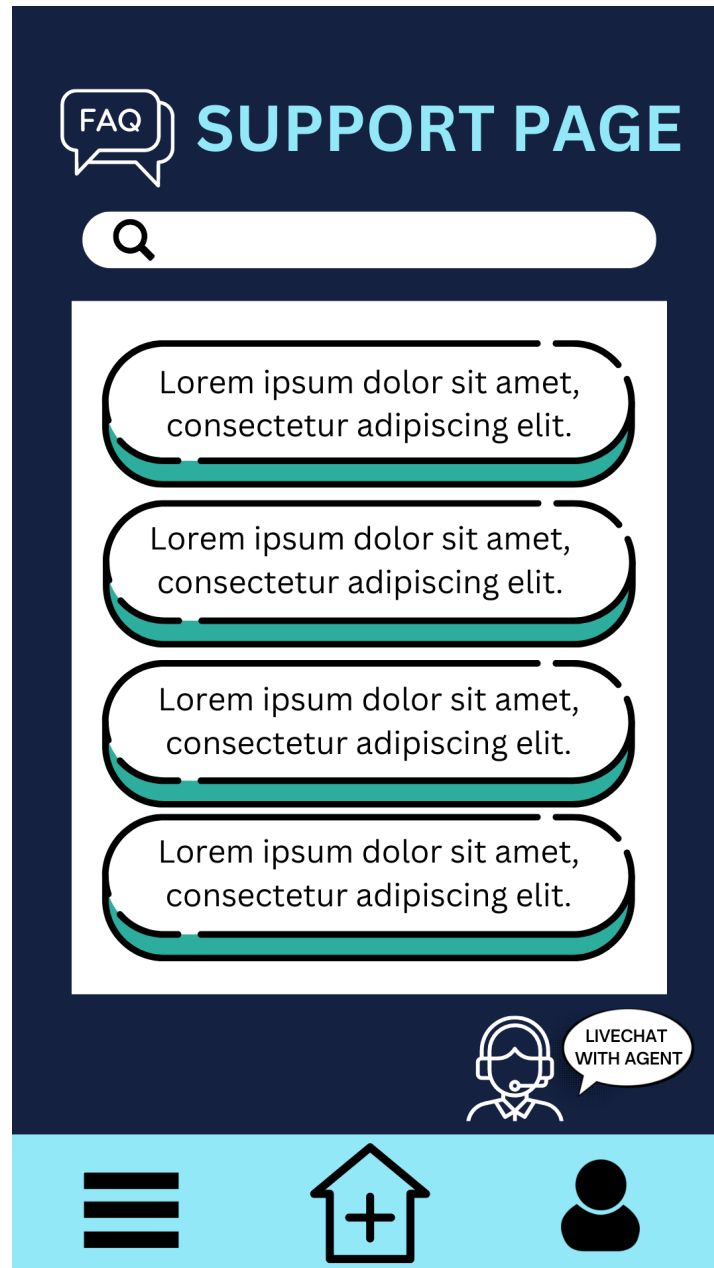
As an Instructor, I want to be able to chat with students

## CHAT (INSTRUCTOR)



As an instructor, I want to be able to contact support to help me with any issues

## SUPPORT



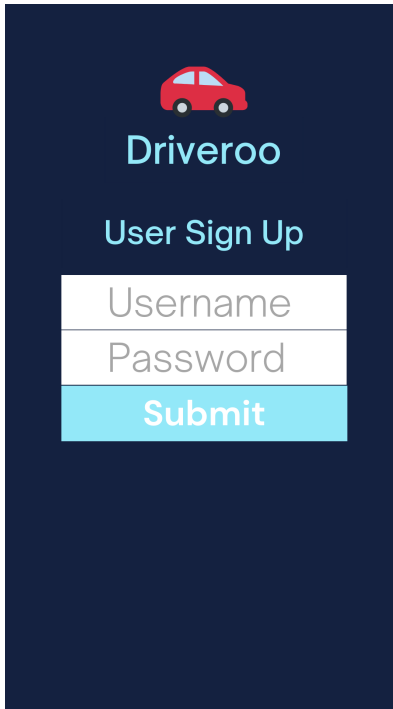
As an Instructor, I want to be able to hold a live-chat with customer service

## SUPPORT LIVE CHAT



## EXTRA SKETCHES WHILE USING THE APP

### USER SIGN UP




A mobile app sketch for the 'User Sign Up' screen. The background is dark blue. At the top center is a red car icon. Below it is the text 'Driveroo' in light blue. Underneath is 'User Sign Up' in light blue. There are three input fields: 'Username', 'Password', and a light blue 'Submit' button. The 'Submit' button is positioned below the 'Password' field.

### MENU FOR USER




## MY LESSONS




### MY LESSONS




**Appointment 1**  
ThisDay 9am-11am  
\$100 via PayPal



**Appointment 2**  
ThatDay 9am-11am  
\$100 via PayPal



## INSTRUCTOR SIGN UP

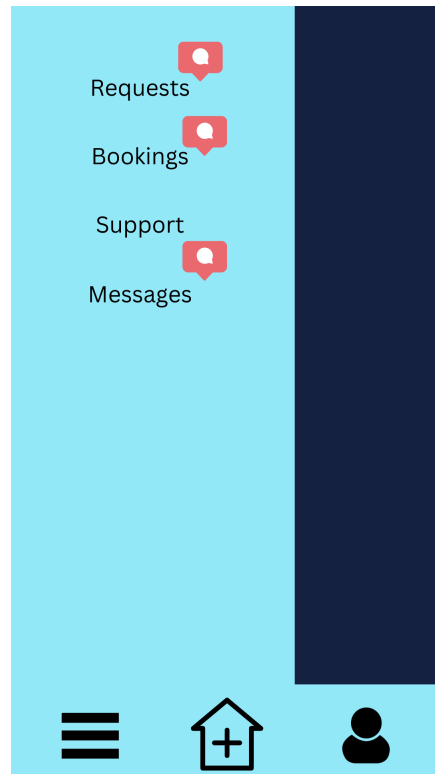


### Driveroo

#### Driver Sign Up

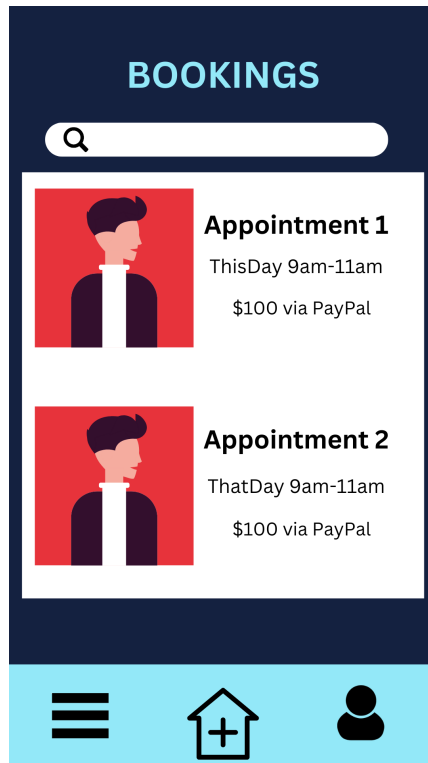
Submit

## INSTRUCTOR MENU



## INSTRUCTOR BOOKINGS





## APPENDIX B (TEST PLAN)

### 1.0 Introduction

#### 1.0.1 Goals

Summarize the testing goals for the project.

#### 1.0.2 Assumptions

Any assumptions which may affect the understanding or execution of this plan should be recorded here.

#### 1.0.3 Risks And Assets

Describe the elements (software or hardware) that are not part of your application but still may impact its correctness and must be checked.

Describe the elements that might positively influence testing on the project.

## **2.0 Scope**

### **2.0.1 Features To Be Tested**

Describe the features and functions that will be tested during the project. This should include functional and non-functional requirements.

### **2.0.2 Features Not To Be Tested**

Describe the features that will not be tested and reason why.

## **3.0 Testing Procedures**

Describe the testing procedures that the project will use. This includes the test lifecycle, types of testing, test objectives, and test criteria.

### **3.0.1 Test Objectives**

Describe the objectives of the testing process.

### **3.0.2 Types Of Testing**

Describe the types of testing that the project will use.

#### **3.0.2.1 Unit Testing**

Describe the strategy for unit testing of the individual subsystems. This includes an indication of the subsystems that will undergo unit tests or the criteria to be used to select subsystems for unit test. Test cases are NOT included here.

### **3.0.2.2 Integration Testing**

Specify the integration testing strategy used. Describe the tests that will be performed in order to verify the interfaces between the subsystems of the software system. This section includes a discussion of the order of integration of subsystems. Test cases are NOT included here.

### **3.0.2.3 Acceptance Testing**

Specify the strategy for testing the software once it has been deployed. This section includes a discussion of the order of acceptance by software function. Test cases are NOT included here.

### **3.0.2.4 Stress Testing**

Identify the limits under which the program is expected to perform (memory constraints, disk space constraints, etc).

### **3.0.2.5 Performance Testing**

Refer to the functional requirements that specify acceptable performance.

## **3.0.3 Testing Tools**

Describe the tools that you will use for testing.

## **4.0 Schedule and Deliverables**

Describe the test deliverables that will be created during the project lifecycle.

Include two tables, one for the schedule of tasks, another for the list of deliverables:

- Acceptance test

- Unit test
- System/Integration test
- Stress test
- Performance test
- Screen prototypes
- Defect reports and summaries
- Test logs and reports

Describe the reports that will be generated by the testing process.

Examples include:

Test Summary Report - A final report of the testing results from the project. Can include items such as total number of test cases, number of test cases executed, % test cases passed, etc.

## **APPENDIX C (END-USER & ADMINISTRATOR MANUALS)**

In this section, include a user manual for your system/application. The user manual should include the following items:

1. Instructions on how to install and configure your system/application, documenting all external software dependencies that need to be setup manually.
2. A user guide for the administrator (use screen shots of your system/application and briefly discuss each screen shot).
3. A user guide for the normal user (use screen shots of your system/application and briefly discuss each screen shot).

## APPENDIX D (PROGRESS MONITORING)

Your team is required to report two items related to progress monitoring in this appendix. The first item is a table summarizing progress and changes during a release with supporting discussion; a sample is shown in [Table 5](#). Notice in Table 5 that all iterations are shown per Release<sup>5</sup>. Also, see *Table 1* in the *Measuring and Monitoring Progress* deliverable.

Table 5: Progress and changes for all iterations [1].

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Story points at start of iteration	130	113	78	31
Completed during iteration	45	47	48	31
Changed estimates	10	4	-3	
Story points from new stories	18	8	4	
Story points at end of iteration	113	78	31	0

The second item is an iteration burndown chart (see [Figure 9](#)) reflecting the data from Table 5.

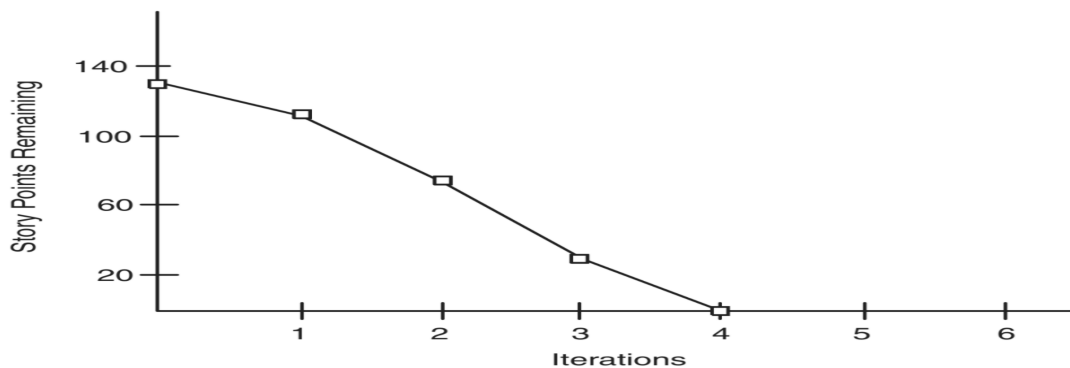


Figure 9: Iteration burndown chart for data from [Table 5](#).

<sup>5</sup> For subsequent Releases, do NOT restart numbering the Iteration. For example, let us assume that we have another Release (i.e., Release 2.0), we would continue numbering our Iterations as *Iteration 5*, *Iteration 6*, and so on.

